Energy, Mines and Resources Canada

Énergie, Mines et Ressources Canada

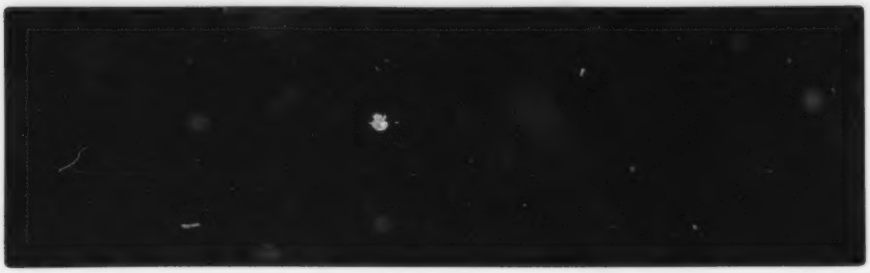
CANMET

Canada Centre for Mineral and Energy Technology

Centre canadien de la technologie des minéraux et de l'énergie

Coal Research Laboratories

Laboratoires de recherche sur le charbon



Canada

AN IMPLEMENTATION OF THE 2 DIMENSIONAL LERCHS-GROSSMAN ALGORITHM

N.J. Stuart

Division Report CRL 89-15(TR)
January 1989

MICROMEDIA

18 pp

MICROMEDIA

AN IMPLEMENTATION OF THE 2 DIMENSIONAL LERCHS-GROSSMAN ALGORITHM

by

N.J. Stuart*

ABSTRACT

A published algorithm is implemented in Pascal. Its ability to find the most profitable pit in a 2 dimensional case of a block model is demonstrated with some simple cases. A larger example is also presented as an illustration.

*Research Scientist, Surface Mining Laboratory, Coal Research Laboratories, Department of Energy Mines and Resources Canada, CANMET, P.O. Bag 1280, Devon, Alberta, TOC 1EO.

APPLICATION DE L'ALGORITHME BIDIMENSIONNEL DE LERCHS-GROSSMAN

N.J. Stuart*

Résumé

Un algorithme a été rédigé en Pascal. Sa capacité à déterminer la mine la plus profitable dans une situation bidimensionnelle d'un modèle par blocs est démontrée à partir de quelques exemples simples. Un exemple plus complexe est aussi présenté à titre d'illustration.

---

*Chercheur scientifique, Laboratoire d'exploitation à ciel ouvert, Laboratoires de recherche sur le charbon, Energie, Mines et Ressources Canada, CANMET, B.P. 1280, Devon (Alberta), T0C 1E0.

CONTENTS

	Page
ABSTRACT	i
RESUME	iii
INTRODUCTION	1
THE ROUTINE	1
THE IMPLEMENTATION	2
A SIMPLE EXAMPLE	2
CONCLUDING REMARKS	4
REFERENCE	4
APPENDIX "A"	5
APPENDIX "B"	10

INTRODUCTION

The Lerchs-Grossman Algorithm is concerned with selecting the optimum pit from a block model. Each block is assigned a value i.e. the profit obtained from the mineral content minus the cost of mining and processing. Thus waste blocks have negative values. Such models can be derived from kridged mineral grades. This algorithm assumes that pit slopes can have a maximum slope of one in one in terms of blocks i.e. the slope angle is determined by the block shape or vice versa. It is a true optimization routine as no better solution can be found, unlike floating cone algorithms.

There is interest in this and other similar algorithms in the industry. This implementation contains all the basic logic of the original paper for the 2 dimensional case.

THE ROUTINE

The program works as follows:-

1) The value data is entered into a 2 dimensional array. This is called tableau 1.

2) Tableau 2 is created by summing the value of each block with all the values of cells directly above it in tableau 1.

3) Tableau 3 is created as follows. Starting at the left side of the model each column is updated in turn. Column 1 stays the same and is largely ignored. The tableau value for that block is added to the tableau three value for that block in the preceding column that:
   a) is either on the same level, one above or one below.
   b) gives the maximum value.

4) The selection of which block used in (3) is stored as an arrow value in a separate array.

5) The maximum value of the pit is the maximum value in level one of the blocks e.g. block (1, m).

6) The shape of the pit can be found by adding the "arrow" value. Suppose the value in arrow (1, m) is 1 one then proceeds to block

arrow (2,m-1) assuming block indices increase to the right and downwards.

THE IMPLEMENTATION

The system stores block values as reals (numbers with a fractional component) but displays them as integers. The value of the pit is presented as a real, however.

The program lists the values in all three tableau and draws the optimal pit with hyphens for unmined blocks and X's for excavated ones. Should the solution pit touch the side of the model a warning is printed as it is possible in this case that the optimum pit extends outside the block model presented. There is no file system in the program as it is for demonstration only.

It should be noted that an extra row (row zero) is required in the arrays to take into account the possibility of not mining that column.

A SIMPLE EXAMPLE

Consider the following block model:

0	0	0	0	0	0	0
-1	-1	-1	-1	-1	-1	-1
-1	-1	2	2	2	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1

A simple approach is to look for economic blocks. However, each of pay blocks with a value of 2 requires three waste blocks (value 3 x -1) to be removed from above it. Thus no block is individually profitable.

-1	-1	-1
	2	

If a cone is fitted as per a floating cone method one finds a cone with a value of zero.

```
      -1    -1    -1    -1    -1
            2     2     2
                  -1
```

How then does our program behave? The block model is tableau one.

Summing the columns to get tableau two:

```
 0     0     0     0     0     0     0
-1    -1    -1    -1    -1    -1    -1
-2    -2     1     1     1    -2    -2
-3    -3     0     0     0    -3    -3
-4    -4    -1    -1    -1    -4    -4
```

And finally tableau three (note the first values are the arrow values):

```
0 : ( 0)   0 ( 0) 0  ( 0) 0  ( 0) 0  ( 0) 0  ( 1) 0  ( 1) 1
1 : ( 0)  -1 (-1)-1  (-1)-1  ( 1)-1  ( 1) 0  ( 1) 1  ( 0) 0
2 : ( 0)  -2 (-1)-3  (-1) 0  ( 0) 1  ( 0) 2  ( 0) 0  (-1)-1
3 : ( 0)  -3 (-1)-5  (-1)-3  (-1) 0  (-1) 1  (-1)-1  (-1)-3
4 : ( 0)  -4 (-1)-7  (-1)-6  (-1)-4  (-1)-1  (-1)-3  (-1)-5
```

This gives the correct optimum pit:

```
-   X   X   X   X   X   -
-   -   X   X   X   -   -
-   -   -   -   -   -   -
-   -   -   -   -   -   -
```

CONCLUDING REMARKS

This is an example of an optimal pit routine i.e. for the given assumptions no better (more valuable) solution can exist than the one found. The appendices contain a program listing and examples of the programs use.

REFERENCE

1. Lerchs, Helmut and I.F. Grossman. "Optimum design of open pit mines". C.I.M. Bulletin, V. 58, No. 633, January, 1965, pp. 47-54.

APPENDIX "A"

PROGRAM LISTING

PROGRAM LISTING

NOTES

This program is written using the Borland International "Turbo Pascal V4". It is, however, standard Pascal except that most other implementations require input and output to be declared, i.e. the first line would be:

PROGRAM lg (input, output);

The improved version of the program with proper in code documentation, right hand margin checking and an improved user interface will be available soon. The constants xmcor and xmrow are set to the maximum depth and width of the model respectively in lines 3 and 4. These can be increased if required beyond the 80 and 40 values shown to a limit defined by your computer/pascal compiler. Larger models would, however, require a reworked i/o routine to display the results.

```
program lg;
const
xmcol=80;
xmrow=40;
var
tab:array[0..xmrow,1..xmcol] of real;
arrow:array[0..xmrow,1..xmcol] of integer;
c,ccount,rcount,mcol,mrow:integer;
v:real;

procedure inp;
begin
ccount:=0;
repeat
      ccount:=ccount+1;
      rcount:=0;
      repeat
            rcount:=rcount+1;
            writeln('enter value for row ',rcount,' and col ',ccount);
            readln(tab[rcount,ccount]);
      until rcount=mrow;
until ccount=mcol;
end;

procedure show;
var i:integer;
begin
writeln;
rcount:=-1;
repeat
      rcount:=rcount+1;
      write(rcount,' : ');
      ccount:=0;
      repeat
            ccount:=ccount+1;
            i:=trunc(tab[rcount,ccount]);
            write(i:3);
      until ccount=mcol;
      writeln;
until rcount=mrow;
writeln;writeln;
end;

procedure show2;
begin
writeln;
rcount:=-1;
repeat
      rcount:=rcount+1;
      write(rcount,' : ');
      ccount:=0;
      repeat
            ccount:=ccount+1;
            write(' (',arrow[rcount,ccount]:2,')',trunc(tab[rcount,ccount]):2);
      until ccount=mcol;
      writeln;
until rcount=mrow;
writeln;writeln;
end;
```

```
procedure t1;
begin
ccount:=0;
repeat
      tab[0,ccount]:=0;
      ccount:=ccount+1;
      rcount:=0;
      repeat
            rcount:=rcount+1;
            tab[rcount,ccount]:=tab[rcount,ccount]+tab[rcount-1,ccount];
      until rcount=mrow;
until ccount=mcol;
end;

procedure t2;
var val:real;
begin
ccount:=1;
repeat
      ccount:=ccount+1;
      rcount:=-1;
      repeat
            rcount:=rcount+1;
            val:=tab[rcount,ccount];
            if rcount>0 then
            begin
                 tab[rcount,ccount]:=val+tab[rcount-1,ccount-1];
                 arrow[rcount,ccount]:=-1;
            end;
            if (tab[rcount,ccount]<=(tab[rcount,ccount-1]+val))then
               begin
                 tab[rcount,ccount]:=tab[rcount,ccount-1]+val;
                 arrow[rcount,ccount]:=0;
               end;
            if ((rcount<mrow) and (tab[rcount,ccount]<=(tab[rcount+1,ccount-1]+val))) then
               begin
                  tab[rcount,ccount]:=tab[rcount+1,ccount-1]+val;
                  arrow[rcount,ccount]:=1;
               end;
      until rcount=mrow;
until ccount=mcol;
end;

procedure getmax;
var i:integer;
begin
v:=0.0;
c:=0;
for i := 1 to mcol do
    begin
        if (tab[1,i]>v) then
        begin
          v:=tab[1,i];
          c:=i;
        end;
    end;
end;
procedure bleckmark;
var i,r,a:integer;
begin
```

```
r:=1;
repeat
   a:=arrow[r,c];
   for i:=1 to r do arrow[i,c]:=2;
   c:=c-1;
   r:=r+a;
   if(r>c)then writeln(' solution failed -need more data to left of model ');
until((r>c)or(c=0));
end;

procedure mine;
var r,c:integer;
begin
r:=0;
repeat
      c:=0;
      r:=r+1;
      repeat
            c:=c+1;
            If (arrow[r,c]=2) then write('X') else write ('-');
      until c= mcol;
      writeln;
until r=mrow;
end;

procedure zero;
var i:integer;
begin
for i:=1 to mcol do tab[0,i]:=0.0;
for i:=0 to mrow do arrow[i,1]:=0;
end;

begin
     writeln('Enter width of model -up to ',xmcol);
     readln(mcol);
     writeln('Enter depth of model -up to',xmrow);
     readln(mrow);

     inp;
     zero;
     show;
     readln;
     t1;
     show;
     readln;
     t2;
     show2;
     readln;
     getmax;
     writeln('maximum value is ',v);
     writeln('righthand margin of pit is',c);
     writeln;writeln;
     blockmark;
     mine;
end.
```

APPENDIX 'B'

ADDITIONAL DEMONSTRATIONS OF THE PROGRAM

Additional demonstrations of the Program.

1) Three blocks only, Only one is 'profitable' taken in isolation, however it is not part of the optimal pit. The blocks with value 14 both each show a profit of -1 if mined on their own. The block with value 6 shows a profit of +1. Thus other algorithms would select this block. However, the l-g algorithm correctly finds the optimum pit is that of removing the two 14 value blocks onle for a profit of 2.

Block Model (Tableau 1):

```
0 :  0  0  0  0  0  0  0  0  0  0  0  0  0
1 : -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
2 : -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
3 : -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
4 : -1 -1 -1 -1 14 -1 -1 -1 14 -1 -1 -1 -1
5 : -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
6 : -1 -1 -1 -1 -1 -1  6 -1 -1 -1 -1 -1 -1
7 : -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

Tableau 2:

```
0 :  0  0  0  0  0  0  0  0  0  0  0  0  0
1 : -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
2 : -2 -2 -2 -2 -2 -2 -2 -2 -2 -2 -2 -2 -2
3 : -3 -3 -3 -3 -3 -3 -3 -3 -3 -3 -3 -3 -3
4 : -4 -4 -4 -4 11 -4 -4 -4 11 -4 -4 -4 -4
5 : -5 -5 -5 -5 10 -5 -5 -5 10 -5 -5 -5 -5
6 : -6 -6 -6 -6  9 -6  1 -6  9 -6 -6 -6 -6
7 : -7 -7 -7 -7  8 -7  0 -7  8 -7 -7 -7 -7
```

Tableau 3:

```
0 :  ( 0) 0 ( 0) 0 ( 0) 0 ( 0) 0 ( 0) 0 ( 0) 0 ( 0) 0 ( 0) 0 ( 0) 0 ( 0) 0 ( 0)
0 ( 0) 0 ( 1) 2
1 :  ( 0)-1 (-1)-1 (-1)-1 (-1)-1 (-1)-1 (-1)-1 (-1)-1 ( 1)-1 (-1)-1 (-1)-1 (-1)-
1 ( 1) 2 ( 0) 1
2 :  ( 0)-2 (-1)-3 (-1)-3 (-1)-3 (-1)-3 (-1)-3 ( 1) 0 ( 0)-2 (-1)-3 (-1)-3 ( 1)
3 ( 0) 1 (-1) 0
3 :  ( 0)-3 (-1)-5 (-1)-6 (-1)-6 (-1)-6 ( 1) 2 ( 0)-1 (-1)-3 (-1)-5 ( 1) 5 ( 0)
2 (-1) 0 (-1)-2
4 :  ( 0)-4 (-1)-7 (-1)-9 (-1)-10 (-1) 5 ( 0) 1 (-1)-2 (-1)-5 (-1) 8 ( 0) 4 (-1)
 1 (-1)-2 (-1)-4
5 :  ( 0)-5 (-1)-9 (-1)-12 (-1)-14 (-1) 0 (-1) 0 (-1)-4 ( 1)-4 ( 0) 6 (-1) 3 (-1
)-1 (-1)-4 (-1)-7
6 :  ( 0)-6 (-1)-11 (-1)-15 (-1)-18 (-1)-5 (-1)-6 (-1) 1 ( 0)-5 (-1) 5 (-1) 0 (-
1)-3 (-1)-7 (-1)-10
7 :  ( 0)-7 (-1)-13 (-1)-18 (-1)-22 (-1)-10 (-1)-12 (-1)-6 (-1)-6 (-1) 3 (-1)-2
(-1)-7 (-1)-10 (-1)-14
```

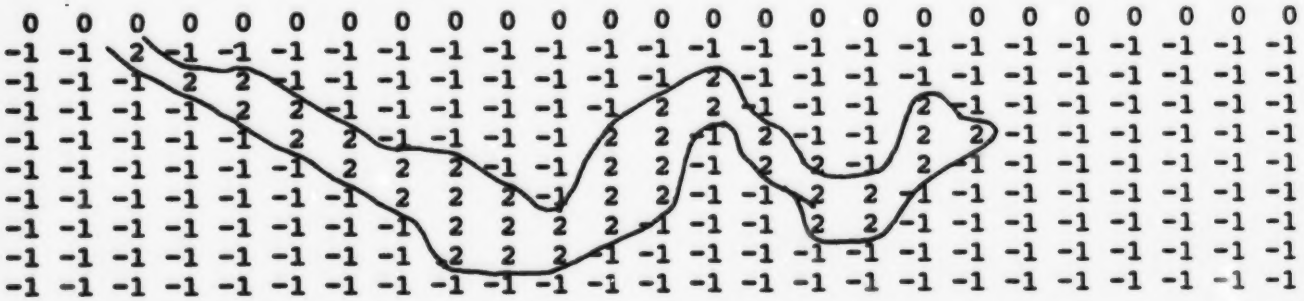
Value and Optimum Pit:

```
maximum value is  2.0000000000E+00
righthand margin of pit is12

-XXXXXXXXXX-
--XXXXXXXX--
---XXX-XXX---
----X---X----
------------
------------
------------
```

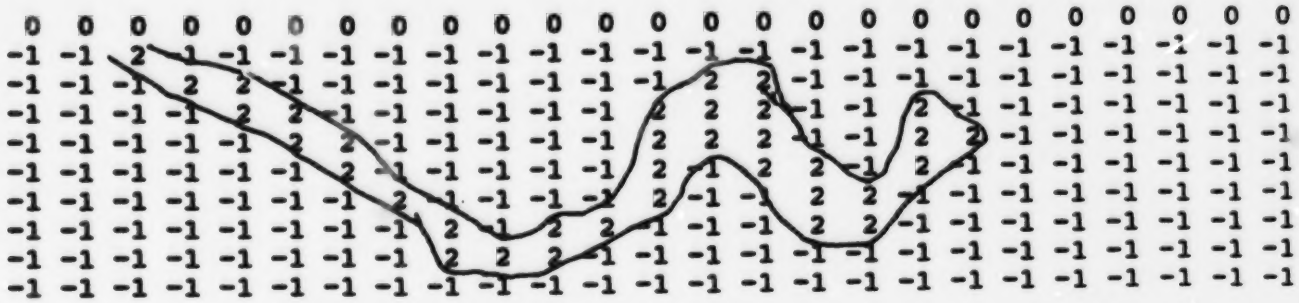
2)A More Realistic Model:

Finally consider these three similar folded structures and their very different optimal pit:-

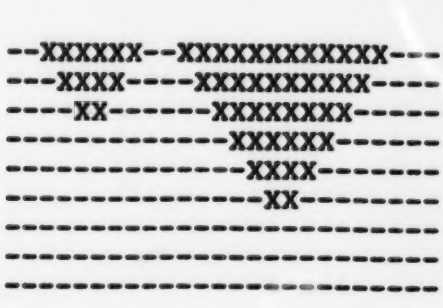


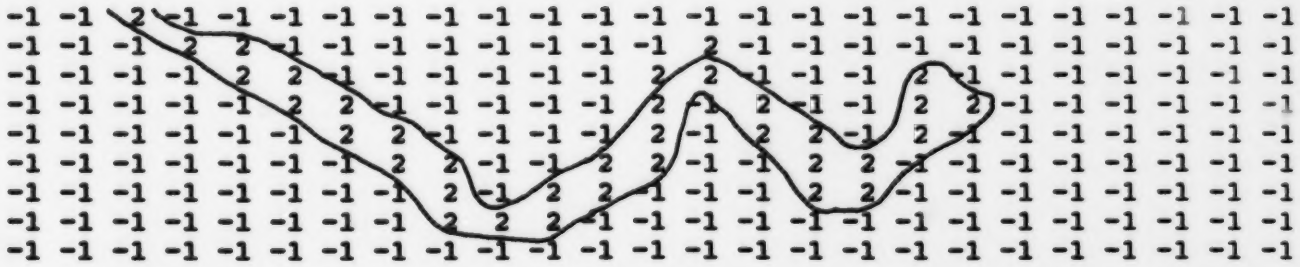

```
maximum value is  1.8000000000E+01
righthand margin of pit is22

--XXXXXXXXXXXXXXXXXXXX---
---XXXXXXXXXXXXXXXXXX----
----XXXXXXXXXXXXXXXX-----
-----XXXXXXXXXXXXXX------
------XXXXXXXXXXXX-------
-------XXXXXX-XXX--------
--------XXXX---X---------
---------XX-------------
-------------------------
enter 1 to repeat 0 to quit
```



maximum value is 6.0000000000E+00
righthand margin of pit is22





maximum value is 3.0000000000E+00
righthand margin of pit is6